SOSS Community Day North America 2024

Critical Conversation: Consuming Open Source Securely

Katherine Druckman and Ryan Ware

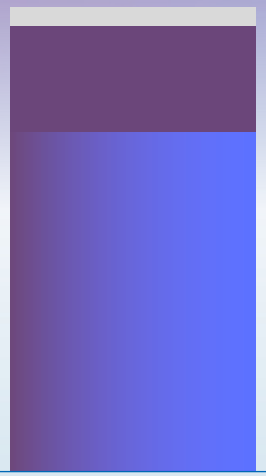
Security challenges

Why is open source security so challenging?

Open source is *everywhere*

96%

of codebases



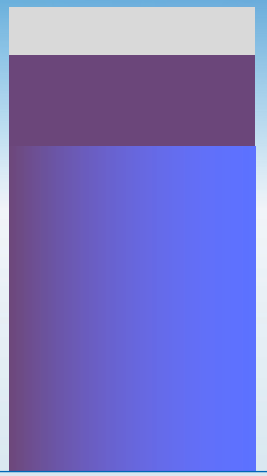
77%

of code within

Source: Synopsis

70–90%

of all software



Source: Linux Foundation

https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html

https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on

Open source is *everywhere*

Yay! We won!

SO MANY PACKAGES

npmjs.org

3,639,251 packages
41,815,290 versions
822,231 maintainers
222,295 namespaces
742,169 keywords
237,045,471,901 downloads

proxy.golang.org

1,105,378 packages
9,970,233 versions
449,953 namespaces
70,056 keywords

hub.docker.com

1,001,771 packages
10,844,967 versions
411,451 namespaces
1,713 keywords
334,237,037,105 downloads

nuget.org

624,254 packages
7,566,455 versions
85,940 maintainers
129,237 keywords
514,134,790,590 downloads

pypi.org

542,396 packages
5,603,074 versions
229,691 maintainers
193,748 keywords
29,768,771,520 downloads

repo1.maven.org

499,556 packages
11,361,001 versions
66,310 namespaces
31,287 keywords

Statistics			
Registries: 59	Versions: 97,334,410	Maintainers: 1,440,484	Keywords: 1,521,288
Packages: 8,840,726	Namespaces: 1,328,127	Downloads: 1,811,443,044,372	

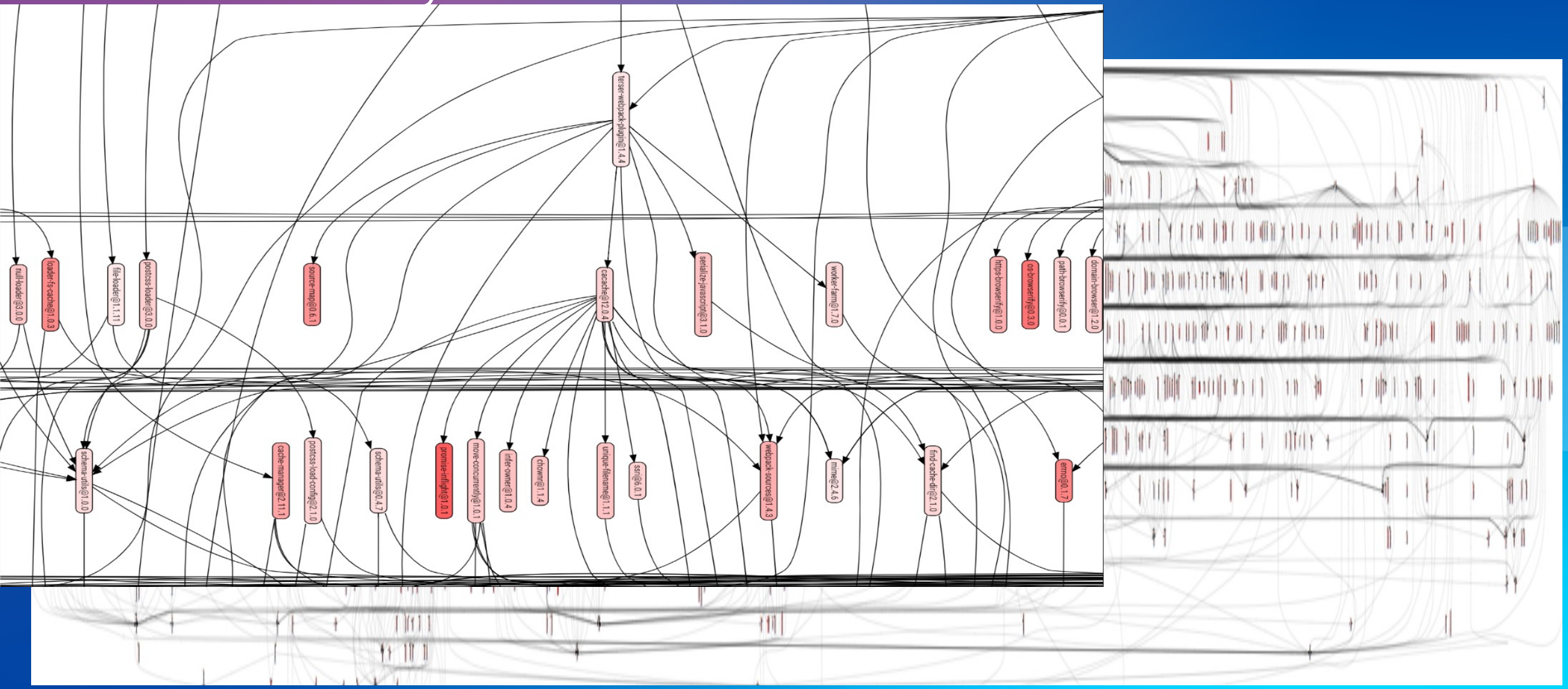
CVEs per year

https://www.cve.org/About/Metrics

So many dependencies

- Secondary and tertiary dependencies can get well into the 100s...
- Especially with web applications

How bad it can get



Evaluating projects

What does it mean to consume open source software securely?

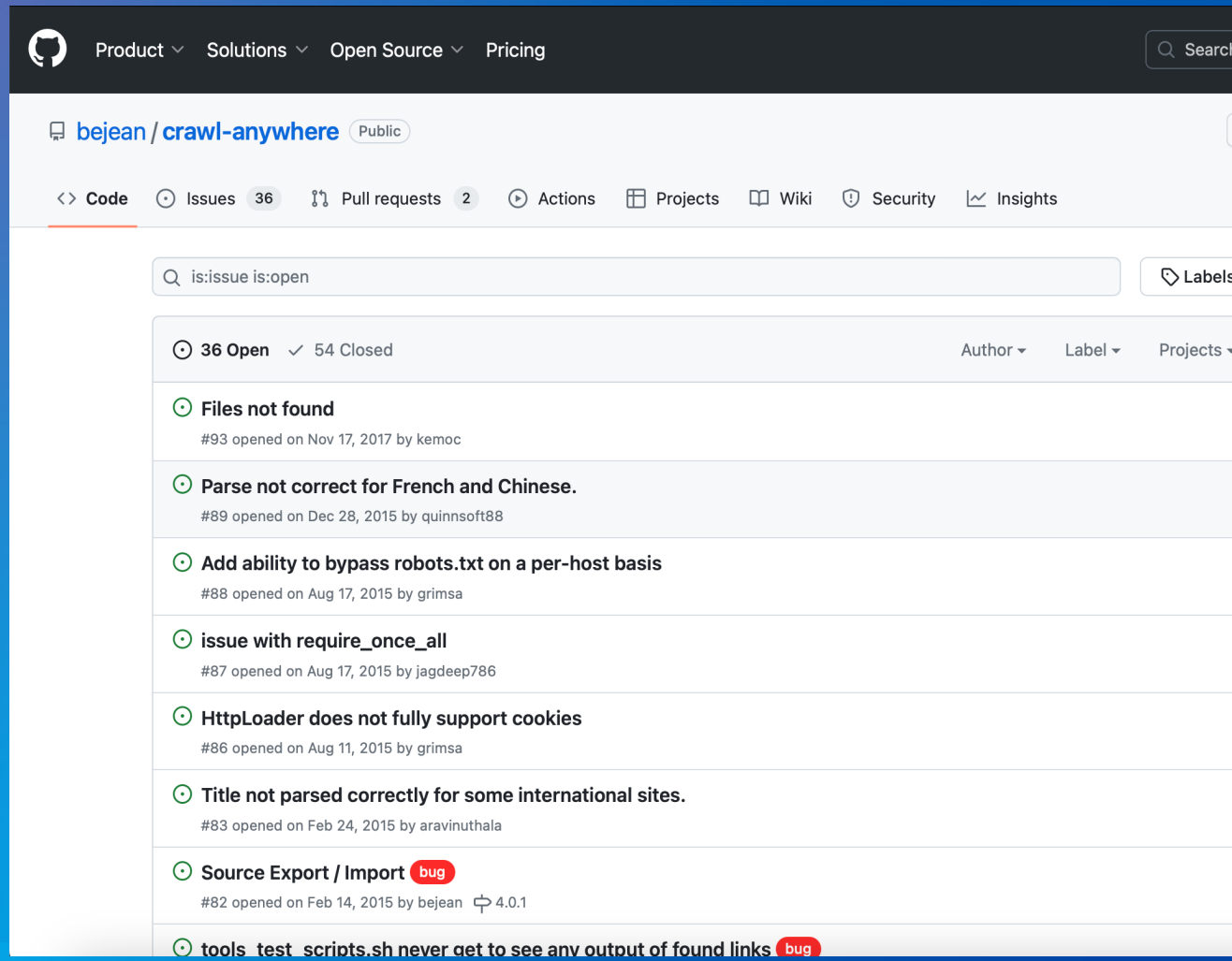
Evaluating open source projects

1. Review basic health—Is it active?
2. Check governance—Is it defined?
3. Review maintenance & releases—Is there a cadence?
4. Explore the community—Are people engaged?
5. Bug reporting—Is there a documented process?

What’s the first thing you would look at when evaluating an open source project to use or include as a dependency?

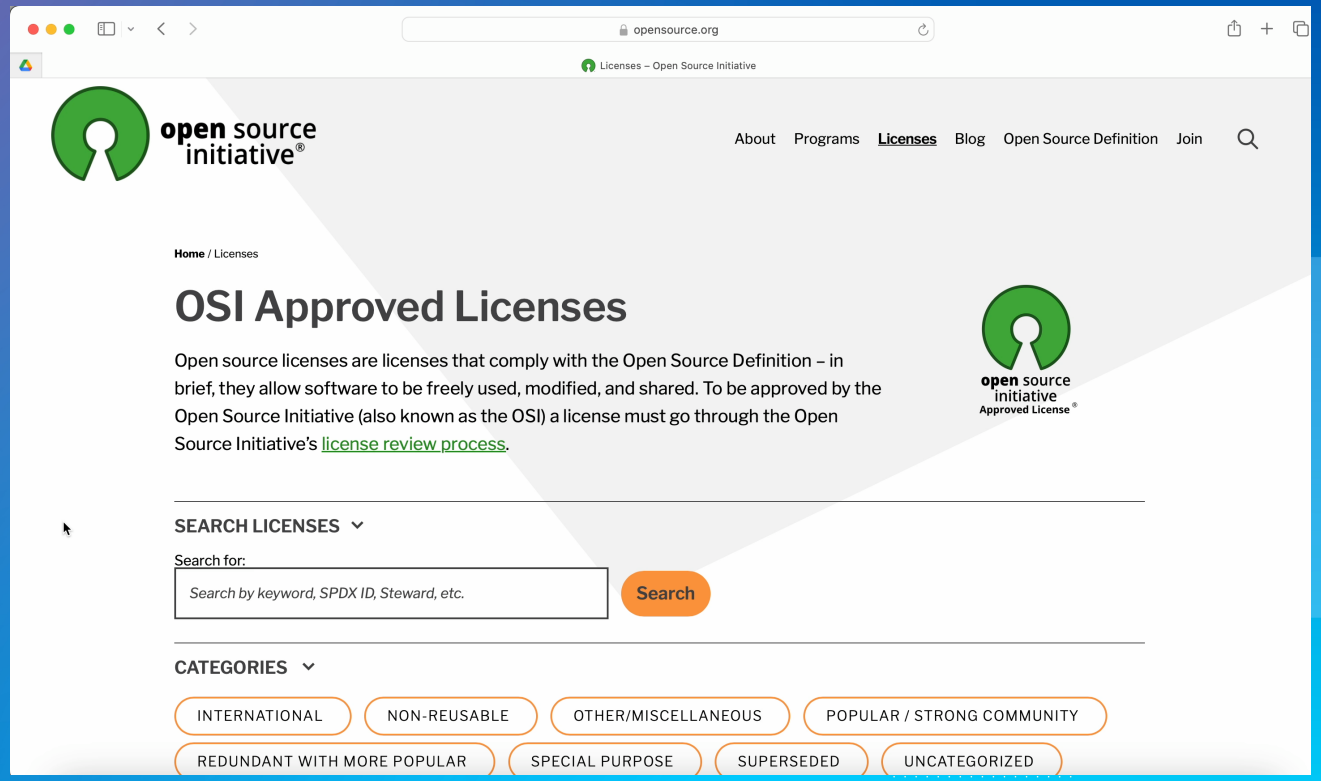
1. Evaluating software: Basic health

- Does the project even have a maintainer anymore?
- When was the last commit?
- Look at the issue queue
  - How active is it?
  - When was the last post?
  - When was the last response to an issue?



2. Evaluating software: Governance

- Clearly defined governance?
  - Clearly stated license? (Hopefully OSI approved)
  - More than one maintainer
  - Maintainers from more than one company or organization
  - How are decisions made?

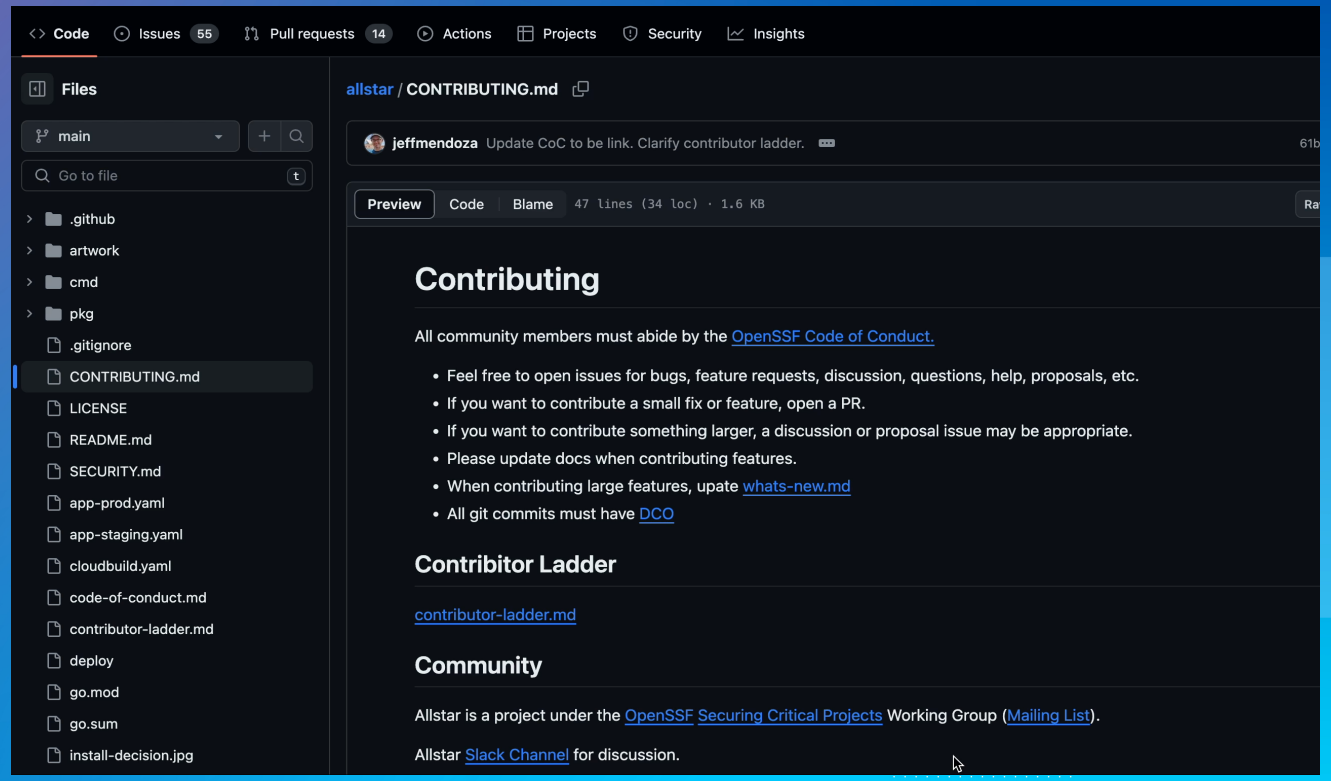


3. Evaluating software: Maintenance and release management

- Has there been substantial activity in the last year?
- Look at the release cadence
  - Is it documented?
  - Regularly occurring?
  - Prompt patch releases to address bugs and security issues?
- Does the project communicate announcements regularly? Does it have a blog?
- Is the latest release a “-alpha” or “-beta,” or does it indicate that it is not yet production-ready?

4. Evaluating software: Community engagement

- Contributor guide?
- Extensively used?
- Is the community working toward security best practices?
  - Automated tests
  - Up-to-date dependencies



5. Evaluating software: Secure bug reporting

Report a Vulnerability

We're extremely grateful for security researchers and users that report vulnerabilities to the Kubernetes Open Source Community. All reports are thoroughly investigated by a set of community volunteers.

To make a report, submit your vulnerability to the Kubernetes bug bounty program. This allows triage and handling of the vulnerability with standardized response times.

You can also email the private security@kubernetes.io list with the security details and the details expected for all Kubernetes bug reports.

You may encrypt your email to this list using the GPG keys of the Security Response Committee members. Encryption using GPG is NOT required to make a disclosure.

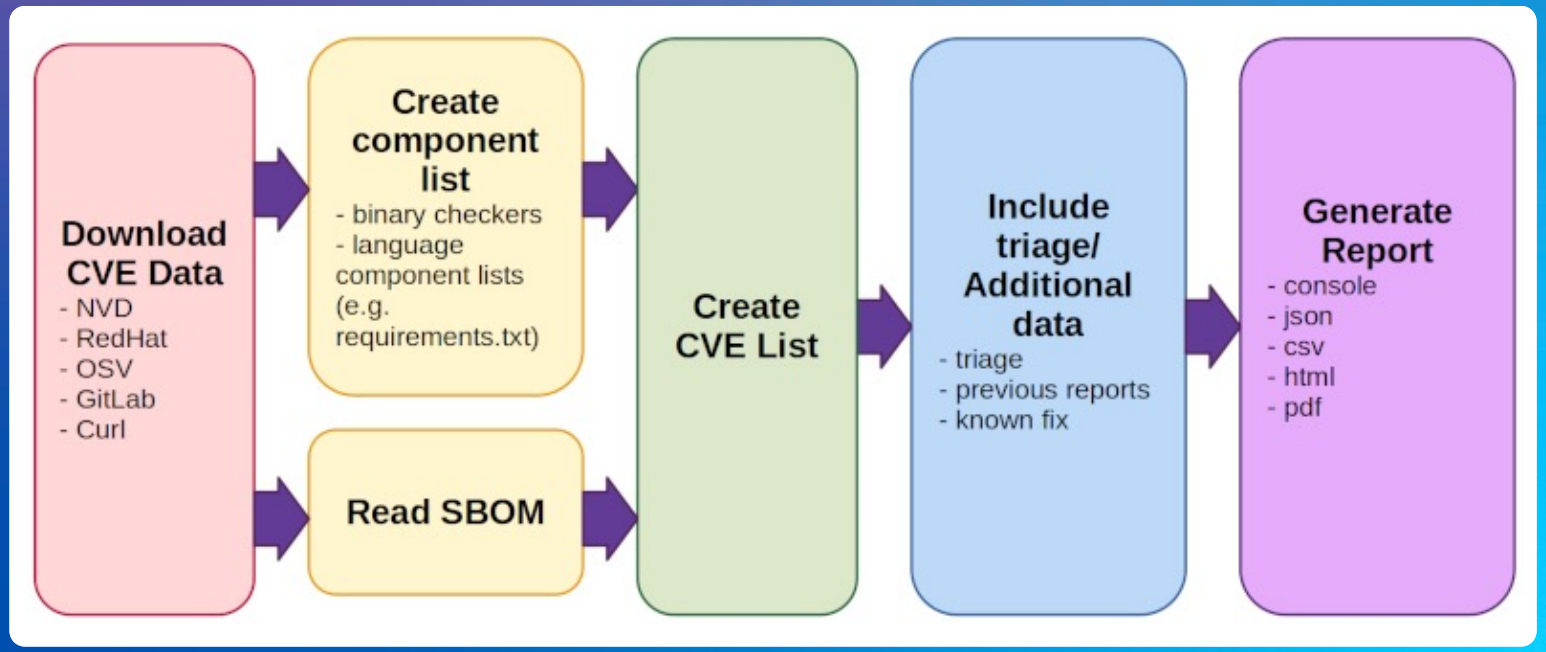
Tools for securing open source software

Beyond basic health

CVE-bin-tool

The CVE Binary Tool can help you find known vulnerabilities in software by using data from the National Vulnerability Database (NVD) list of Common Vulnerabilities and Exposures (CVEs) as well as known vulnerability data from Redhat, Open Source Vulnerability Database (OSV), Gitlab Advisory Database (GAD), and Curl.

1. A binary scanner—Helps you determine which packages may have been included as part of a piece of software.
2. Tools for scanning known component lists—Such as CSV files, SBOM formats, etc.



As simple as ...

pip install cve-bin-tool

cve-bin-tool <directory/file>

OpenSSF Best Practices Badge

README | LGPL-3.0 license | Security

Gramine Library OS with Intel SGX Support

docs passing | openssf best practices passing

A Linux-compatible Library for Multi-Process Applications

What is Gramine?

Gramine (formerly called *Graphen* a lightweight library OS, designed to run a singl host requirements. Gramine can run cations in an isolated environment with benef a complete OS in a virtual machine -- luding guest customization, ease of porting to process migration.

Gramine supports native, unmodified Linux binaries on any platform. Currently, Gramin SGX enclaves on Linux platforms.

In untrusted cloud and edge deployments, there is a strong desire to shield the whole infrastructure. Gramine supports this "lift and shift" paradigm for bringing unmodified Confidential Computing with Intel SGX. Gramine can protect applications from a malici minimal porting effort.

OpenSSF Best Practices 100%

CORE INFRASTRUCTURE INITIATIVE BEST PRACTICES

LibreOffice

Projects that follow the best practices below can voluntarily self-certify they've achieved an Open Source Security Foundation (OpenSSF) best Show details

If this is your project, please show your badge status on your project page! The like this: openssf best practices passing Here is how to embed it: Show details

These are the passing criteria. You can also view the silver or gold level

Expand panels | Show all details | Show only incomplete c

Basics

Identification

What is the human-readable name of the project? Show details

LibreOffice

OpenSSF Scorecard

- What is it?
  - Quick, easy project assessment via list of automated checks for best practices
- What does it help protect me from?
  - Malicious maintainers and packages
  - Poorly maintained projects
  - Compromised build systems and/or code
- How do I use it?
  - Command line interface (CLI)
  - GitHub Action

README | Code of conduct | Apache-2.0 license | Security

OpenSSF Scorecard

openssf scorecard 9.6 | openssf best practices passing | build passing | CodeQL passing | reference | go report A+
codecov 75% | SLSA level 3 | slack openssf/scorecard

Overview

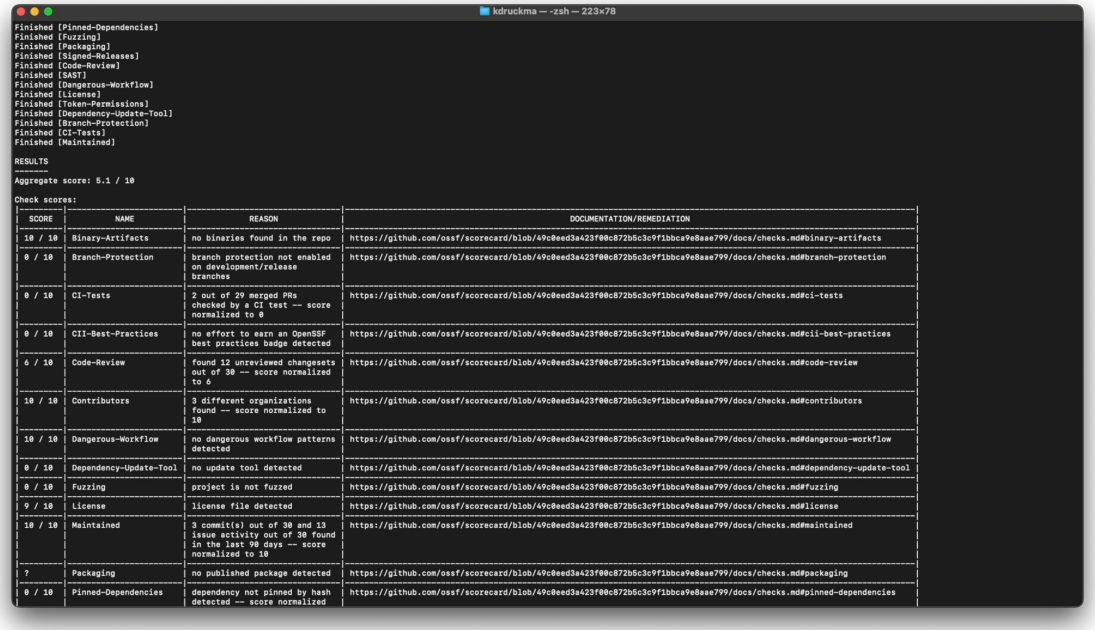
- What Is Scorecard?
- Prominent Scorecard Users
- View a Project's Score
- Scorecard's Public Data

Using Scorecard

- Scorecard GitHub Action
- Scorecard REST API
- Scorecard Badges
- Scorecard Command Line Interface
  - Prerequisites
  - Installation
  - Authentication
  - Basic Usage

OpenSSF Scorecard

Score in terminal …



kdruckma — -zsh — 223×78

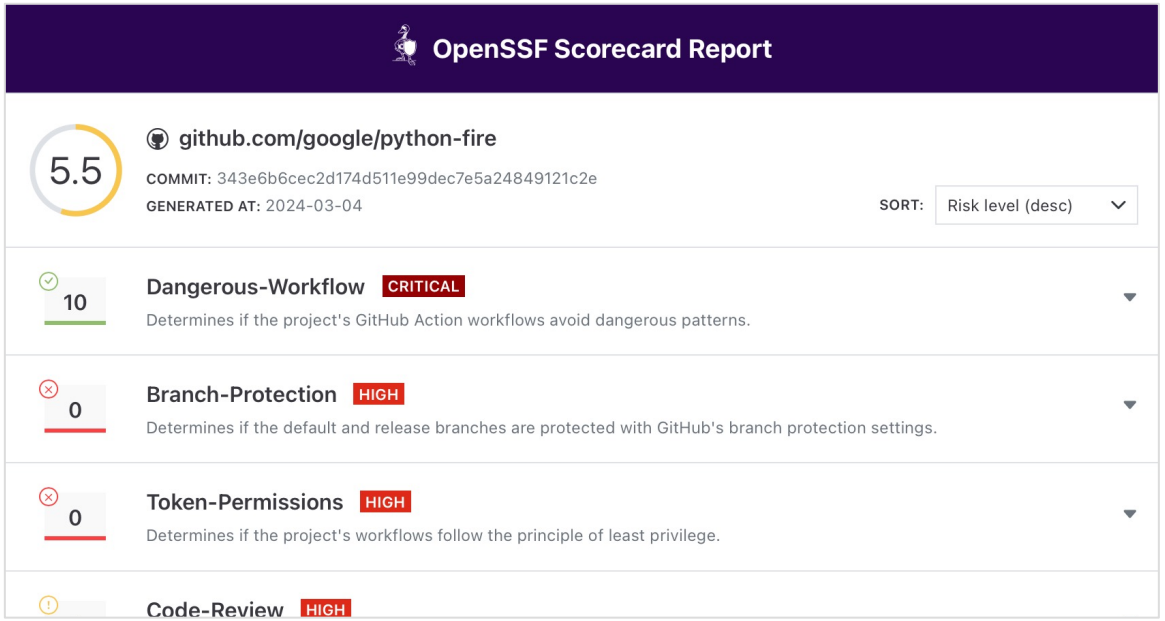
```
Finished [Pinned-Dependencies]
Finished [Fuzzing]
Finished [Packaging]
Finished [Signed-Releases]
Finished [Code-Review]
Finished [SAST]
Finished [Dangerous-Workflow]
Finished [License]
Finished [Token-Permissions]
Finished [Dependency-Update-Tool]
Finished [Branch-Protection]
Finished [CI-Tests]
Finished [Maintained]

RESULTS
-------
Aggregate score: 5.1 / 10

Check scores:
```

SCORE	NAME	REASON	DOCUMENTATION/REMEDIATION
10 / 10	Binary-Artifacts	no binaries found in the repo	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#binary-artifacts
0 / 10	Branch-Protection	branch protection not enabled on development/release branches	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#branch-protection
0 / 10	CI-Tests	2 out of 29 merged PRs checked by a CI test -- score normalized to 0	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#ci-tests
0 / 10	CII-Best-Practices	no effort to earn an OpenSSF best practices badge detected	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#cii-best-practices
6 / 10	Code-Review	found 12 unreviewed changesets out of 30 -- score normalized to 6	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#code-review
10 / 10	Contributors	3 different organizations found -- score normalized to 10	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#contributors
10 / 10	Dangerous-Workflow	no dangerous workflow patterns detected	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#dangerous-workflow
0 / 10	Dependency-Update-Tool	no update tool detected	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#dependency-update-tool
0 / 10	Fuzzing	project is not fuzzed	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#fuzzing
9 / 10	License	license file detected	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#license
10 / 10	Maintained	3 commit(s) out of 30 and 13 issue activity out of 30 found in the last 90 days -- score normalized to 10	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#maintained
?	Packaging	no published package detected	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#packaging
0 / 10	Pinned-Dependencies	dependency not pinned by hash detected -- score normalized	https://github.com/ossf/scorecard/blob/49c0eed3a423f00c872b5c3c9f1bbca9e8aae799/docs/checks.md#pinned-dependencies

… or via browser



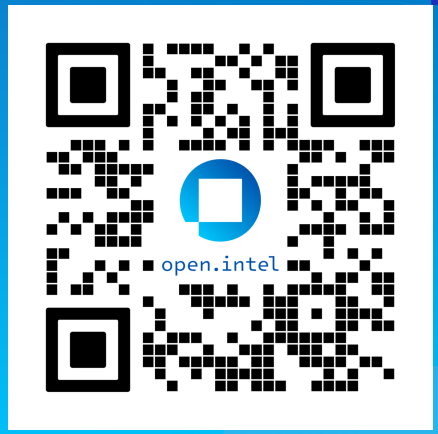
Taking ownership

Developers don’t owe you anything

Scan for tools, sources, and resources

Visit the presentation's GitHub page

- Intel and OpenSSF tools
- Guides and community resources
- Links to articles and source material
- A PDF of this presentation



Where to find us

- Twitter/X: @katherined
- Mastodon: @katherined@reality2.social
- LinkedIn: katherinedruckman
- GitHub: kdruckman

- LinkedIn: ryan-ware
- Mastodon: @ware@infosec.exchange
- GitHub: ware

Notices and disclaimers

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Intel® technologies may require enabled hardware, software, or service activation. No product or component can be absolutely secure. Your costs and results may vary.